METHOD AND APPARATUS FOR SELECTING PARTICULAR DECODER BASED ON BITSTREAM FORMAT DETECTION

This invention relates generally to devices for decoding audio and/or video bit streams of varying format without producing presentation artifacts. This invention relates more

5   particularly to detecting format changes quickly enough to switch decoders before a presentation artifact can be generated.

Digital audio and visual data are sent in bit streams over communication lines, including the internet, in a variety of data compression formats (hereinafter "format"). Each unique format may require a correspondingly unique decoder to convert the bit stream content

10   information into its presentation form: audio or video. Examples of such formats are four MPEG formats ISO/IEC 13818-3, ISO/IEC 11172-3, ISO/IEC 13818-7 and ISO/IEC 14496-3 and the AC-3 format. Stream identifications (IDs) are provided in the formatted data stream which were originally intended to identify the format type. As the standards have evolved, stream IDs have become unreliable because they do not uniquely identify the

15   format of the formatted bit stream. For example, stream ID 0xC0 is used for all four of the MPEG formats while two of them, ISO/IEC 13818-7 and ISO/IEC 11172-3 require different decoders. Both video and audio data is conventionally arranged in frames.

An approach to format detection has been to use independent a priori information about the bit stream as a format guide. The information must be parsed to be used. At modern data

20   rates, such slow methods leave gaps, or presentation artifacts, in the presentation of the data. Even with the best tools available, the exact point at which the format change actually takes place cannot be reliably and precisely indicated. Thus, current methods of parsing a priori independent information to switch content decoders usually still produces annoying presentation artifacts.

25   Thus an apparatus and a method is needed to select content decoders fast enough to avoid the creation of presentation artifacts.

The present invention provides an apparatus for switching between a plurality of bit stream decoders, each decoder adapted to decode bit streams having a unique format. The apparatus comprises one or more format detectors which gather information only from the

30   bit stream. The format detector output feeds logic, which is responsive to the detector output to determine a particular appropriate decoder among the plurality of decoders. The particular decoder determined by the logic may be responsive to the formatted bit stream to decode the formatted bit stream. At least one switch, responsive to the determination of

the logic, connects the formatted bit stream to the input of the particular decoder. The
format detectors detect at least one of the frame information, the packet header
information, and the code word values in a portion of a coded content of the bit stream,
wherein the code word values, derivatives thereof, or a pattern of code word values, are

5   unique to the bit stream format. Methods are disclosed for detecting formats for switching
between decoders for formatted bit streams. The methods may employ a plurality of format
detectors arranged in sequence, parallel, or a hybrid arrangement. In sequenced format
detectors, the later stages may only be used when a previous stage returns an ambiguous
result. The methods comprise detecting the bit stream format only from the bit stream. The

10   method further comprises determining the correct decoder for decoding the detected bit
stream format and switching the formatted bit stream to the correct decoder.
The present invention has the advantage of being faster than table-parsing and other a
priori methods. Only a small amount of data need be processed to determine format and
select the correct decoder.

15   The present invention has the advantage of being adaptable, even as standards proliferate
and confusion among the various standards increases, because any truly unique, new
standard must be detectable by at least one of the methods provided.
The foregoing and other features and advantages of the seamless agile decoder will be
apparent from the following more particular description of specific embodiments of the

20   seamless agile decoder, as illustrated in the accompanying drawings, wherein:
FIG. 1 depicts a flow chart of the steps of an exemplary embodiment of the method of the
present invention;
FIG. 2 depicts a flow chart of the steps of a second exemplary embodiment of the method
of the present invention;

25   FIG. 2A depicts a flow chart of the steps of a third exemplary embodiment of the method
of the present invention;
FIG. 3 depicts a block diagram of an exemplary embodiment of the apparatus of the
present invention;
FIG. 4 depicts a block diagram of a second exemplary embodiment of the apparatus of the

30   present invention; and
FIG. 5 depicts a block diagram of a third exemplary embodiment of the apparatus of the
present invention.

Referring now to the figures, FIG. 1 depicts an exemplary method 100 of detecting the format of a bit stream. At the start 100, a data segment from the formatted data stream may be read for analysis. The data segment will conventionally contain frame synchronization

5   and frame structure information, such as a potentially unique start code, a packet header, and content which includes potentially unique code words. The first step 102 searches the data segment for a particular start code M, which reflects frame synchronization and frame structure information, also called "frame information." The start code indicates the starting point of a frame and may be read directly. Some formats use unique start codes. Upon

10   obtaining the particular start code M, step 103 determines if the start code M uniquely indicates a particular decoder N to be used for the detected data stream. In some embodiments, other frame information may be used. For example, the frame size, which is the number of bits between start codes, may be unique to some formats. A start code is an example of a format indicator. A format indicator is a feature that may be read from the bit

15   stream and from which information about the format of the bit stream may be gathered. "Start code" is used herein to represent format indicators based on frame information. The relationship between formats and decoders is not strictly one-to-one. Decoder N may represent a particular one of a plurality of decoders, which may be selected based upon a particular one start code of a plurality of possible start codes. Start code M represents a

20   particular one of all the possible start codes. The set of the plurality of decoders that may be decoder N may partially overlap the sets of decoders that may be decoder Y and decoder Q. That is, an individual decoder may be selected on the basis of a start code, a packet header, or a code word detection at different times for slightly different formats which may use the same decoder.

25   The determination may be made, for example, by looking up the start code in a start code table contained in or available to the logic 360 (FIG. 3) implementing the decision. For example, start code M may be associated with decoder N in the start code table, so performing a lookup in the table may return a decoder N determination. For a counter-example, if the start code M has no decoder associated with it in the start code table, the

30   lookup operation will not return a decoder determination. The table lookup does not require parsing of the table. For example, the table may be a hash table. If start code M is uniquely associated with decoder N, then the appropriate decoder N has been determined

and may be selected in step 110. When the decoder N is selected, it begins decoding the formatted bit stream. If start code M does not uniquely identify a decoder N, then an ambiguity exists: which decoder to use is unknown.

In step 104, the ambiguity is avoided by using a second approach to uniquely detecting the

5   format: the packet header X may be read and compared to entries in a packet header table. The packet header X may uniquely determine a decoder Y to be used, as indicated by finding decoder Y associated with packet header X in a table lookup. If a decoder Y is uniquely determined, then decoder Y may be selected in step 111. If a decoder Y is not uniquely determined, then an ambiguity still exists.

10   In step 106, the ambiguity is again avoided by using a third approach to uniquely detecting the format: code word P may be read and compared in step 107 to entries in a table to determine if decoder Q is uniquely determined. If code word P does not uniquely identify a particular decoder Q, then the ambiguity remains unresolved. In that case, step 108 reads the next data segment and the process begins again at 101. If code word P does uniquely

15   determine a decoder Q, then decoder Q is selected in step 112. In an alternate embodiment, step 107 may be modified to find a plurality of code words in the data segment, and to match a pattern of such code words to a table entry to uniquely determine a decoder.

Code words may be obtained in different ways. Different code books are used for different

20   coding standards which, in turn, require different decoders. Code books contain the code words used for encoding the data in the contents portion of the data packet. Some codes contain code words unique to their corresponding formats. For example, a Huffman audio coding standard may contain a unique code word. A unique code word determines the format, or coding, and, therefore, the appropriate decoder. In some cases, a unique

25   signature can be derived from a code that does not have unique code words. For example, a Huffman code without unique code words may be decoded to extract the actual audio sample value and that value can be applied to a reconstruction equation unique to each format. If the value is improper for a particular format, it will cause an overflow result from the reconstruction equation. The reconstruction equations for all formats may be

30   tested to find the one that does not overflow. That reconstruction equation determines the format, and therefore, the correct decoder to be used. The most likely formats are tested first. Code word derivatives such as the actual audio sample value are also referred to

herein as "code word values." Video coding uses a variety of coding methods, including sub-band coding, motion-compensation coding, and vector quantization. Each method has a separate code book and may have unique code words. Derivation methods may be devised for each coding method by an approach similar to that discussed above. Patterns of

5   code words may also be unique. For example, several codes may have code words A and B, but only one may have A, B, and C, where C is not independently unique. Deriving code words may be a slow process in some instances, so it is usually the last choice among the detection methods presented.

After any decoder selection 111-112, step 113 decides if repeating confirmation is desired.

10   If so, the next data segment is read in step 108, and the process restarts at step 102. If not the process halts 114 until an external trigger initiates process 100 again at step 101. In some embodiments, periodic format determination 100 may be desired. In other embodiments, continuous format determination may be desirable. For example, if someone attempted to protect data by including frequent format changes, continuous

15   format detection would be needed. In such embodiments, step 113 repeats the format detection at a predetermined interval, which may be every data segment. The order in which format detectors are called begins with the fastest (the start code detector 102-103) and ends with the slowest (code word detector 106-107). Other orderings are contemplated, for example, for embodiments normally detecting data streams that are

20   known to have no unique start codes. Also, other criteria for ordering the steps may be used, such as putting the format detector with the highest probability of determining a unique decoder first, and the format detector with the lowest probability of unique determination last.

The process 100, like the processes described below, may be implemented in hardware,

25   software, firmware, or hybrids thereof. Because adaptability to presently undefined future formats is important, a software implementation may be preferred for some applications. For other applications, hardware embodiments may be preferred for their speed. Embodiment 100 may be the fastest exemplary embodiment depicted. The other embodiments may be attractive for particular applications. For example, in a real-time

30   computer application wherein the accuracy of the determination is critical, the embodiment depicted in FIG. 2 may be useful.

packet header detector 211-212 is never accessed. In an alternate embodiment, all

detectors 202-203, 211-212, and 205-206 are accessed before step 207.

As with embodiment 100, a question of repetition is raised in step 214 after decoder

selection 213, 215, or 216. If repetition is not desired, the process ends at step 215.

5   Otherwise, step 207 acquires the next data segment and the process starts again at step 202.

Those of ordinary skill in the art will appreciate that the format detections 202-203, 211-

212, and 205-206 may be interchangeable as to order. Such variations of embodiment 200

are contemplated in the present invention. Likewise, those of ordinary skill in the art will

appreciate that additional format detections may be added to embodiment 200.

10   Some alternate embodiments may be designed for systems that receive only limited subsets

of all possible formats, and the order of the format detections may be adapted to those

systems.

FIG. 2A depicts an exemplary embodiment 600 of a parallel method for format detection.

Each of the three detection methods 602-603, 611-612, and 621-622, run concurrently.

15   Decisions made by the parallel detection methods 602-603, 611-612 , and 621-622 may be

accumulated and sorted in step 607. If more than one format detection 602-603, 611-612,

and 621-622 each find a unique decoder, decoder selections 604, 613, or 623 may be sorted

according to a criterion, such as first-in-time criterion, in step 607. The first-in-time

detection of a unique decoder results in selection 604, 613, or 623, respectively, of an

20   appropriate decoder being selected in step 607. A first-in-time detection selection 604,

613, or 623 initiates step 605, which determines if the confirmation should be repeated. If

so, step 615 acquires the next data segment, shutting off any processing by the other two

detection methods. If repetition is not desired, the process 600 ends at step 606, also

shutting off the other two detection methods. Either way, a first-in-time unambiguous

25   determination controls the decoder selection. In some embodiments, the first-in-time

selection algorithm in step 606 may be replaced by different selection criteria. For

example, the results of all successful detections may be obtained and subjected to a

probability criterion or a weighted probability criterion in step 606. In some embodiments,

step 606 may be omitted, allowing a first-in-time decoder selection algorithm to control.

30   If none of the detection methods 602-603, 611-612, or 621-622 produce an unambiguous

decoder determination, additional steps 614 and 624 may be implemented. As each

detection method fails, any information gathered by the attempt is used to update a default

ambiguity descriptor in step 614. For example, the ambiguity descriptor may be a binary number, with each digit, or bit, corresponding to a particular decoder. The default value is zero for each bit, indicating no detector has been found. If the first detection method determines that there are three possible decoders, but cannot determine which of the three

5   is correct, each bit in the ambiguity descriptor corresponding to a possible decoder is changed to a one. The next detection method to finish also produces an ambiguity descriptor, which is bitwise logically ANDed with the first ambiguity descriptor. Thus, those decoders that are found to be possible, or cannot be eliminated by both of the first two methods, will each have a one in the bit corresponding to the possible decoder. The

10   information from the third method to produce results is similarly incorporated. Step 624 then asks if a unique decoder "A" has been found, i.e., is there only one bit with a value of one in the ambiguity descriptor? If so, that decoder "A" is selected in step 625. In an embodiment, step 625 may make a first-in-time decoder determination based upon the individually ambiguous outputs of the two fastest methods. Thus, even if the slowest

15   method was destined to arrive at a unique decoder determination, the sum of the two fastest detection outputs may find it first. The advantage of embodiment 600 is that it may use all of the information gathered by the format detections. If repetition is desired, the process returns to step 601.

FIG. 3 shows a block diagram of an exemplary embodiment 300 of the seamless agile

20   decoder apparatus. Formatted data stream 310 may be split to connect to the format detectors 320 and a switch 330. Format detectors 320 include hardware and/or software to instantiate process steps such as 102, 104, 106, 202, 211, and/or 205 or variations thereof (See FIGS. 1-2). The detected start code, packet header, or code word is sent by path 324 to the logic 360, which may uniquely determine which decoder 340, 342, 344 may is

25   appropriate for the format detected. In some embodiments, format detectors 320 and logic 360 are integrated. Once logic 360 has determined the appropriate decoder, switch 330 directs the formatted data stream on line 303 to the selected decoder 340, 342, or 344. Line 303 may include delays or buffers to compensate for the processing time of the format detectors and logic. Decoder 340, 342, or 344, as selected, produces decoded data 350,

30   352, or 354, respectively. In some embodiments, the switch 330 is integral with logic 360 and format detectors 320.

FIG. 4 shows an exemplary alternate embodiment 400 of the seamless agile decoder wherein all three format detectors 420, 422, and 424 each receive copies of a data segment from formatted data stream 410. In some embodiments, the data segment will be the same data segment for all format detectors. In an alternate embodiment, the data segments may

5   be sequential data segments, with the slowest format detector getting the first data segment and the fastest format detector getting the last in the sequence. All detectors 420, 422, and 424 may operate simultaneously.

The outputs of the format detectors 420, 422, and 424 are sent to the association logic 460. Association logic may instantiate two selection methods. The first method is to select the

10   first-in-time unambiguous decoder determination and activate switch 430 to implement that decoder determination. The second method, used when all outputs are ambiguous, is to take the first-in time ambiguity descriptor and refine it with the subsequent ambiguity descriptors from the other format detectors to determine if a single decoder remains after all refinements. If a single decoder is indicated by the final ambiguity descriptor, that

15   decoder is selected. Otherwise, the next data segment is acquired and the process restarts. If the process instantiated in embodiment 400 succeeds, formatted stream data 410 is connected along line 426 to switch 430, and there through to the appropriate decoder 440, 442, or 444. Line 426 may have delays or buffers to compensate for the processing time of detectors 420, 422, and 424, and the association logic 460. In some embodiments, the

20   format detectors, association logic, and the switch may be integral. In many embodiments, the decoders 440, 442, and 444 are also integrated with the detectors 420, 422, and 424, and the association logic 460.

FIG. 5 shows a block diagram of a particular alternate embodiment 500 having a single format detector 520. This embodiment illustrates that, even with a single format detector

25   520, logic 560 is still required to determine the appropriate decoder from the code word, packet header, or start code detected. Logic 560 includes lookup tables for determining a decoder 540, 544, or 548 from the detected code word, packet header, or start code. Switch 530 may be a microelectronic circuit integral to the logic 560.

As formats evolve, it is possible that a single format detector 520 will suffice. It is also

30   contemplated that other detectible format features may be discovered, perhaps as patterns of code words, start codes, and packet headers, or that new detectible format features may

be added in the future. The detection methods provided herein are extensible to new and additional detectible format features.

The foregoing description has described selected embodiments of methods and apparatuses for seamless agile decoding.

5   While the invention has been particularly shown and described with reference to selected embodiments thereof, it will be readily understood by one of ordinary skill in the art that, as limited only by the appended claims, various changes in form and details may be made therein without departing from the spirit and scope of the invention. For example, a seamless agile decoder may comprise a plurality of format detectors for different frame

10   synchronization and frame structure features.